A review of game tree pruning

Jessica Löhr

Universität Heidelberg

jessica-loehr@web.de

2. May, 2019

Overview

Thomas Antony Marsland

- A review of game tree pruning (1986)
- Canadian computer scientist
- Co-creator of Principal Variation Search
- Creator of computer chess program “Wita”
- Participated at computer chess championships

Pruning

What is pruning?

Methods to decrease number of nodes in a game tree that need to be evaluated

Pruning

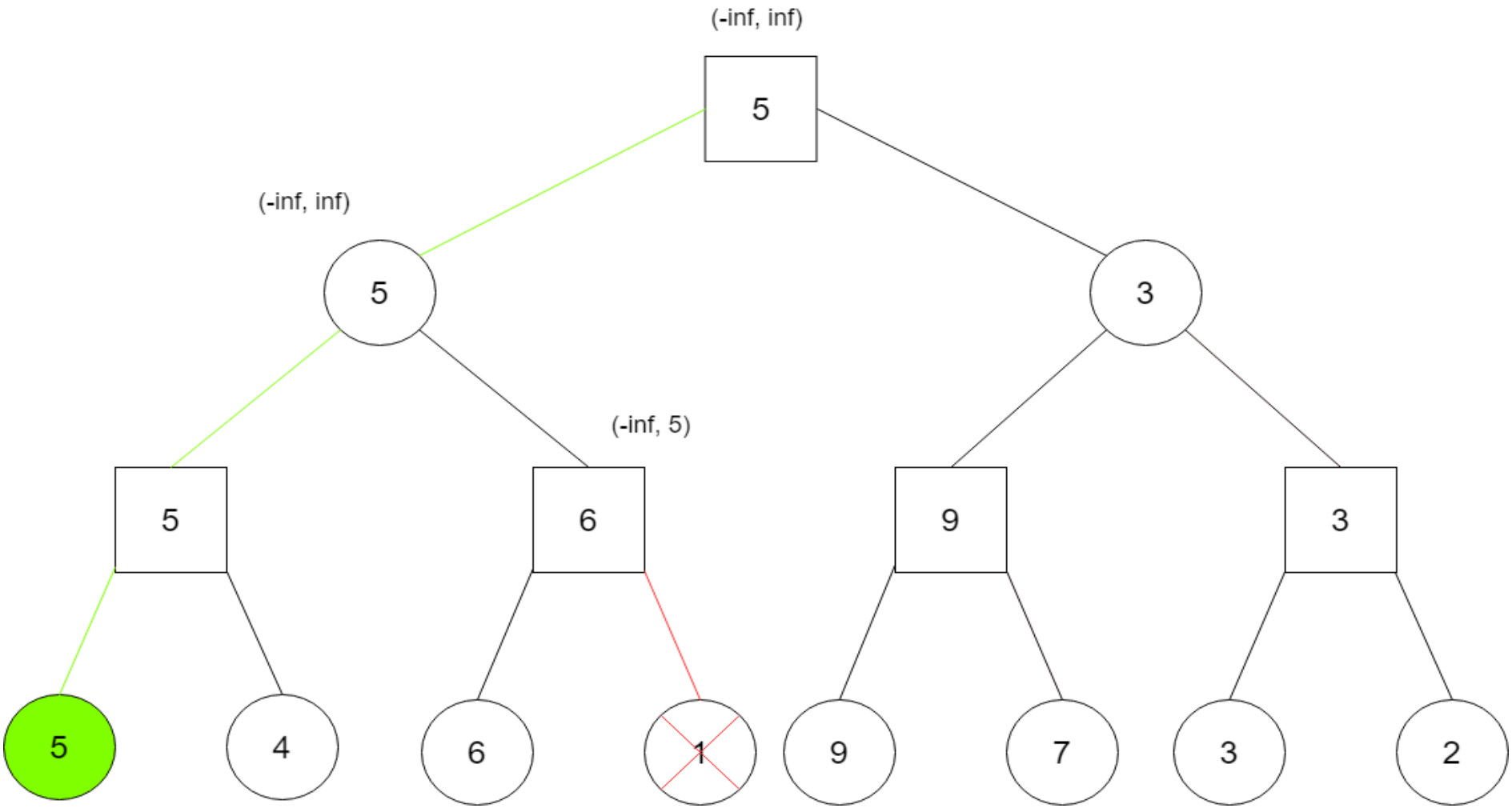
What is pruning?

Methods to decrease number of nodes in a game tree that need to be evaluated

Alpha-Beta-Pruning:

- function alphabeta(node, depth, α, β)
- $\alpha =$ current best move/ lower bound
- $\beta =$ upper bound
- Returns evaluation function if depth $==$ 0

Alpha-Beta-Pruning



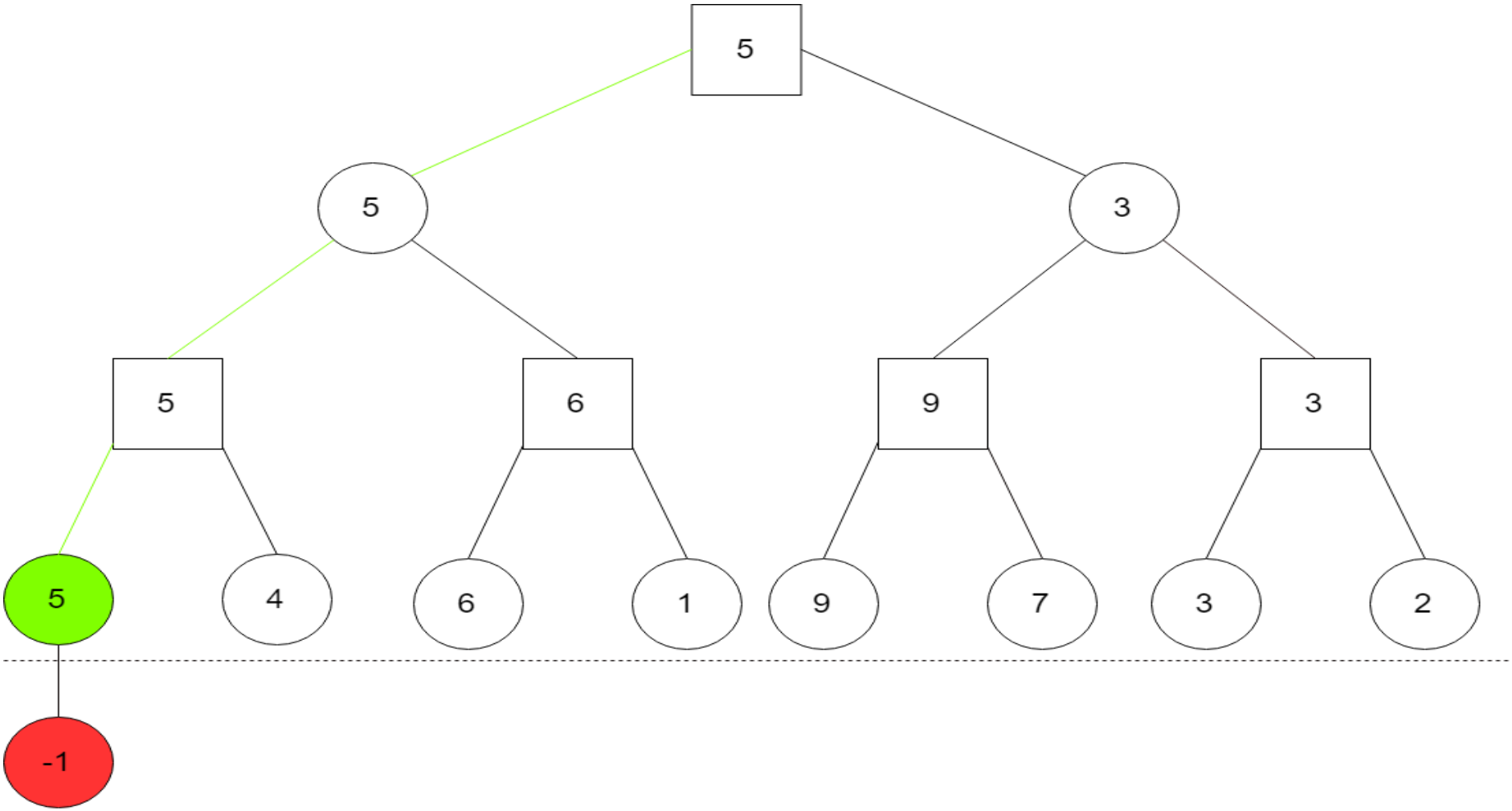
Overview

In this talk we are going to...

- ...improve alpha-beta pruning
- ...discuss more methods and problems

Do the methods still hold up today?

Horizon Effect



Quiescence Search

Horizon Effect

How can we avoid making moves that are refuted after the maximum search depth?

Solution:

- Increasing search depth for certain moves
- Search until position gets ‘quiet’

Further Enhancements

Performance

How can we prune more branches?

Idea:

- Change $\alpha - \beta$ bounds
- Aspiration Search

Aspiration Search

How to choose the bounds?

- Standard Alpha-Beta: α = -inf, β = inf
- Estimate score V of position
- Use material for estimation

Piece	Value
pawn	1
knight	3
bishop	3
rook	5
queen	9

- Choose error limit e
- $\alpha = V - e$, $\beta = V + e$
- Re-search if score is out of bounds

Aspiration Search

How to choose the bounds?

- Standard Alpha-Beta: α = -inf, β = inf
- Estimate score V of position
- Use material for estimation

Piece	Value
pawn	1
knight	3
bishop	3
rook	5
queen	9

- Choose error limit e
- $\alpha = V - e$, $\beta = V + e$
- Re-search if score is out of bounds

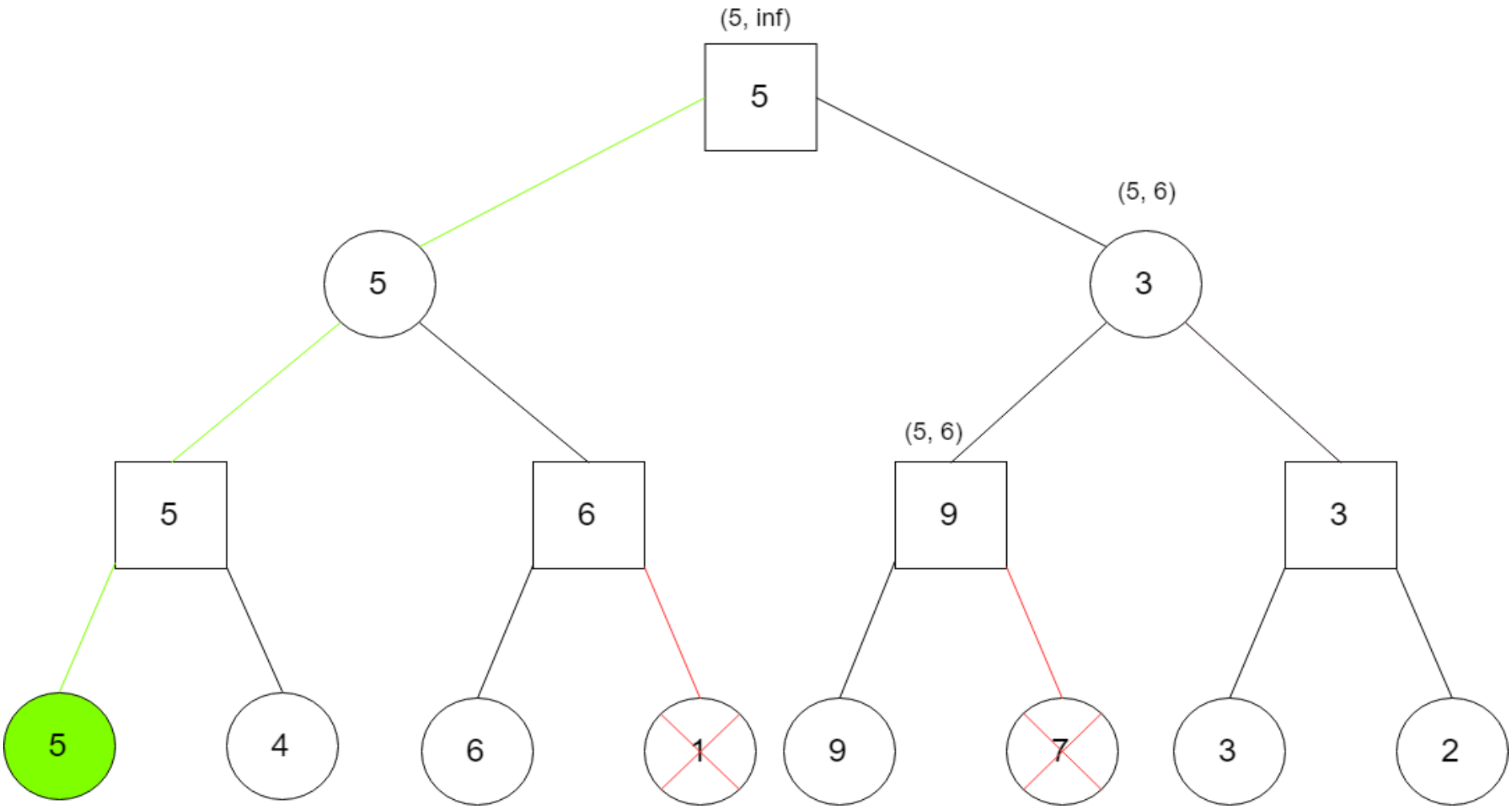
Does not work well in positions with complex captures

Principal Variation Search

How to improve the search method?

- Assumption: moves are ordered from best to worst
  $\rightarrow$ first move as α bound
- Search further moves with bounds α and $\alpha + 1$
  - score $< \alpha$: worse move
  - score $> \beta$: β cutoff
  - $\alpha <$ score $< \beta$: re-search branch

Principal Variation Search



Principal Variation Search

- More branches pruned
- Branches might be re-searched
- Works best if moves are ordered

Move Ordering

Move Ordering

Sort moves so that most plausible ones are searched first (often capture moves)

Killer heuristic:

- Saves moves that refute a line
- Two most frequent moves per depth

Move Ordering

Move Ordering

Sort moves so that most plausible ones are searched first (often capture moves)

Killer heuristic:

- Saves moves that refute a line
- Two most frequent moves per depth

History heuristic:

- Table of size 64x64
- Frequency for every move is stored

Transposition Tables

Positions might be re-visited
$\rightarrow$ Use of tables

Transposition Tables

Positions might be re-visited
$\rightarrow$ Use of tables

Can be used to:

- Narrow α - β bounds
- Move re-ordering
- Look up score of subtrees

Transposition Tables

Possible entry:

Entry	Explanation
Move	Best move in position
Score	Value of subtree
Flag	Tree fully searched?
Height	Depth of subtree upon score is based

Conclusion

Many ways to improve standard alpha-beta-pruning:

- Quiescence Search
- Principal Variation Search
- Use of tables

Methods are still used by modern chess engines!

Stockfish

- Stockfish 10: one of todays best chess engines
- Open source
- https://stockfishchess.org/

Thanks for your attention!